

**Appl. No. 10/640,620**  
**Amdt. dated December 9, 2008**  
**Reply to final Office action of September 10, 2008**

**Amendments to the Specification:**

Please replace the section headed "Related Applications" with the following amended section.

**RELATED APPLICATIONS**

The present application is related to the following commonly owned U.S. Patent Applications:

U.S. Patent Application entitled "INSTRUMENTING JAVA CODE BY MODIFYING BYTECODES," filed concurrently herewith under Attorney Docket No. 10017135-1 U.S. Patent Application Serial No. 10/640,626;

U.S. Patent Application entitled "USE OF THREAD-LOCAL STORAGE TO PROPAGATE APPLICATION CONTEXT IN JAVA 2 ENTERPRISE EDITION (J2EE) APPLICATIONS," filed concurrently herewith under Attorney Docket No. 200311221-1 U.S. Patent Application Serial No. 10/640,619;

U.S. Patent Application entitled "PROPAGATING WEB TRANSACTION CONTEXT INTO COMMON OBJECT MODEL (COM) BUSINESS LOGIC COMPONENTS," filed concurrently herewith under Attorney Docket No. 10017133-1 U.S. Patent Application Serial No. 10/640,625; and

U.S. Patent Application entitled "SYNTHESIZING APPLICATION RESPONSE MEASUREMENT (ARM) INSTRUMENTATION," filed concurrently herewith under Attorney Docket No. 10017138-1 U.S. Patent Application Serial No. 10/640,623.

Please replace the section headed "Brief Description of the Drawings" with the following amended section.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

FIGURE 1 schematically depicts an exemplary distributed multi-tier Web application architecture in which transaction monitoring agents according to the teachings of the invention are incorporated.

FIGURE 2 schematically depicts the use of an instrumentation engine according to one embodiment of the invention for modifying bytecode representation of selected methods of a class.

FIGURE 3 schematically depicts the use of another instrumentation engine according to another embodiment of the invention for modifying bytecode representation of selected methods of a class.

FIGURE 4 illustrates a prototype of an exemplary hookclass method, which can provide a portion of the HookControl interface shown in FIGURES 2 and 3, according to one embodiment of the present invention.

FIGURE 5 illustrates a prototype of a hookMethod method, which can provide a portion of the HookControl interface shown in FIGURES 2 and 3, according to one embodiment of the present invention.

FIGURES 6A-C contain a flowchart illustrating how classes and methods can be selected for instrumentation by the control object of FIGURES 2 and 3, according to one embodiment of the present invention.

FIGURE 7 is a listing of an exemplary wrapper method that can be produced by the instrumentation tools of FIGURES 2 and 3.

FIGURES 8A-C contains a flowchart illustrating how a method can be instrumented by the instrumentation tools of FIGURES 2 and 3, according to a simplified embodiment of the invention.

FIGURE 9 illustrates prototypes of exemplary classLoadStart, defMethod and classLOadEnd methods ~~that can be used the plug-in instrument objects of FIGURES 2 and 3~~, according to one embodiment of the present invention.

FIGURE 10 illustrates prototypes of exemplary methodEntry, reportkg and methodExit methods ~~that can be used the plug-in instrument objects of FIGURES 2 and 3~~, according to one embodiment of the present invention.

FIGURE 11 illustrates prototypes of exemplary methodEntryOneArg and methodException methods ~~that can be used the plug-in instrument objects of FIGURES 2 and 3~~, according to one embodiment of the present invention.

FIGURE 12 illustrates a prototype of an exemplary getHook method ~~that can be used the plug-in instrument objects of FIGURES 2 and 3~~, according to one embodiment of the present invention.

FIGURES 13A-D contain a listing of an exemplary null plug-in instrument class, such as one that could be used with the ExecCallback interfaces of FIGURES 2 or 3, according to one embodiment of the present invention.

FIGURE 14 schematically illustrates an ARM interface in communication with an application and management agents.

FIGURE 15 schematically depicts the interaction of an instrumented class having hooks for communicating with an ARM agent for invoking ARM calls.

FIGURE 16 schematically depicts an ARM agent in communication with a measurement server to transmitting records corresponding response time of transactions thereto.

FIGURE 17 schematically depicts storing correlators corresponding to transactions in a hierarchical child-parent transaction chain in Java thread local storage stack.

FIGURE 18 depicts exemplary correlators for parent-child transactions shown in FIGURE 17

FIGURE 19 depicts an exemplary record generated by an ARM agent for one of the transactions depicted in FIGURE 17.

FIGURE 20 illustrates an exemplary multi-tier web application architecture having, among other elements, a browser and a web server on which monitoring agents according to the teachings of the invention are deployed.

FIGURE 21 schematically illustrates a multi-tier web architecture in which a web server, upon the receipt of an HTTP request from a browser, generates an Active Server Page.

FIGURE 22 schematically depicts an exemplary COM object implementing three interfaces.

FIGURE 23 schematically depicts a wrapper COM object corresponding to the COM object of FIGURE 22.

FIGURE 24 an exemplary method utilized by the wrapper COM object of FIGURE 23 to call invoke ARM calls and to call the original method of the wrapped object.

FIGURE 25 schematically depicts utilizing a hook to load a selected dynamic link library upon invocation of selected system functions by an object to be wrapped.

FIGURE 26 schematically depicts patching the code of a selected system function, namely, CoCreateLInstance by inserting a jump instruction therein.

FIGURE 27 schematically depicts instructions in a function provided by the loaded dynamic link library of FIGURE 25 for generating a wrapper object.

FIGURE 28 depicts a chain of COM objects in a parent-child hierarchical relation.

FIGURE 29 schematically depicts passing correlators among COM objects operating on different platforms.